**Outline**

Access the Python Development environment and follow the tutorial to gain an initial exposure to a programming language. Begin to develop an familiarity with basic programming concepts.

**Objectives**

* Use correct terminology to describe programming concepts;
* Describe the types of data that computers can process and store (e.g., numbers, text);
* Explain the difference between constants and variables used in programming;
* Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

* Python3 Development Environment at: //repl.it/
* Python Tutorial at: <http://www.letslearnpython.com/learn/>

**Accessing the Python3 Web IDE Environment**

Accessing the IDE

* Go to: <https://repl.it/>
* Select Python3
* Sign-up / Create an account
* Make sure you can remember your account information for the rest of the course.

Using the IDE

* Use the black area like a calculator to try simple statements or commands
* Use the white area to create programs with multiple statements

**Accessing the Tutorial**

Accessing the Tutorial

* Go to: <http://www.letslearnpython.com/learn/>
* Read up to “Lesson 3: Math”

**Level 1: Basic Math & Strings**

Access the Tutorial and start at “Lesson 3: Math”.

Questions

1. Complete “Lesson 3: Math – Math Basics” by typing the sample commands in the black area of the IDE.
   1. Create your own expression using 5 “+” and “-“ operators.
   2. List your expression and the result below.

5+32-5 = 32

1. Complete “Lesson 3: Math – More Operators” by typing the sample commands in the black area of the IDE.
   1. Create your own expression using 5 “\*” and “/” operators.
   2. List your expression and the result below.

5\*3 = 15

1. Complete “Lesson 3: Math – More Division” by typing the sample commands in the black area of the IDE.
   1. Create one division expression that gives a whole number answer

10/2 = 5

* 1. And one division expression that gives a decimal number answer.

10/4= 2.5

* 1. List your expressions and the results below.

1. Complete “Lesson 3: Math – Floats” by typing the sample commands in the black area of the IDE.
   1. Use the “round()” function for the expressions you created in question #3 above.
   2. List your “round()” expressions and the results they return below.

“Round (10/4)” = 2

1. Read through “Lesson 3: Math – Comparison Operators”.
   1. Why do you think Equals is “==” instead of “=”?

When you put in “comparison operators ” python turns it into a question so if you put in 5>=6, the program reads it as a question “ five is equal to or greater then 6?”

* 1. What does “=” mean?

Equal means is this the same to something else for example 4+5 = 9 its used in math.

1. Complete “Lesson 3: Math – Practice” and “Lesson 3: Math – Practice Answers” by typing the sample commands in the black area of the IDE.
   1. Create an expression using 5 different operators that returns a “True” result

5<=9

4<=5

8<=11

1>=0

2>=1

* 1. And an expression using 5 different operators that returns a “False” result.

5>=9

4>=5

8>=11

1<=0

2>=1

* 1. List your expressions and the results returned below.

5<=9 true

4<=5 true

8<=11 true

1>=0 true

2>=1 true

5>=9 false

4>=5 false

8>=11 false

1<=0 false

2>=1 false

1. Complete “Lesson 4: Strings – Strings” and “Lesson 4: Strings – Examples” by typing the sample commands in the black area of the IDE.
   1. Explain why typing “apple” works and why typing apple without quotes gives an error.

-

When you type apple with quotes it registers everything inside as words and not as a math problem

* 1. Also explain why “2 + 5” does not equal 7.

Because since 2+5 Is inside quotes it is not registered as a math problem.

1. Complete “Lesson 4: Strings – Operators” by typing the sample commands in the black area of the IDE.
   1. Explain why typing “appl” + “e” works and why typing “apple” - “e” gives an error.

Typing “appl” + “e” works because the python treats it as two separate words, since there’s an addition sign it’ll form into one word. Typing “apple” – “e” doesn’t work because it comes as an unsupported operand

* 1. Also explain why “Hello” \* 10 works but why “Hello” / 10 does work.

You can’t divide the word hello 10 ways because its not a number and hello only has 5 letters in it but you can multiply the word to repeat itself for example “hello” \* 2 would equal hello hello.

1. Complete “Lesson 4: Strings – Indexes” by typing the sample commands in the black area of the IDE.
   1. List the letters in your first name and the index for each letter in your first name.

G-u-r-m-a-n

0-1-2-3-4-

1. Complete “Lesson 4: Strings – Indexes Examples” by typing the sample commands in the black area of the IDE.
   1. Explain why print(“Hello!”[4]) does not print “l”.

Because the index starts at 0 then goes on to 1,2,3. So the correct answer wil be ‘o’

* 1. What does print(“Hay, Bob!”[4]) print? For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5])

This will print the space because everything inside the quotes will register as a index even the space.

1. Complete “Lesson 4: Strings – Rules” by typing the sample commands in the black area of the IDE.
   1. Explain why print(“Hello!”[7]) gives an error.

Because there are not seven letters in “Hello!” so there’s no index place for 7.

**Level 2: Booleans & Variables**

Access the Tutorial and start at “Lesson 5: Variables”

Questions

1. Complete “Lesson 5: Variables – Save a Value” by typing the sample commands in the black area of the IDE.
   1. What do you get if you type puppies / 3?

Traceback (most recent call last): File "python", line 1, in <module> NameError: name 'puppies' is not defined

* 1. Why doesn’t typing kittens / 3 work?

Because you cant divide the word kittens

1. Complete “Lesson 5: Variables – Assign a New Value” by typing the sample commands in the black area of the IDE.
   1. Explain how the following sequence of commands works:
      * puppies = 36

you are assigning the word puppies the value of 36

* + - puppies = puppies / 6
    - puppies

1. Read through “Lesson 5: Variables – Rules”.
2. Complete “Lesson 5: Variables – Math Operators” by typing the sample commands in the black area of the IDE.
   1. Explain what happens for following sequence of commands:

colour = “red”

=> None

* + - puppies = 36

now when you type puppies it will give you back 36

* + - colour + puppies

1. Traceback (most recent call last):
2. File "python", line 1, in <module>
3. NameError: name 'colors' is not defined
4. Traceback (most recent call last):
5. File "python", line 1, in <module>
6. NameError: name 'colors' is not defined
7. Traceback (most recent call last):
8. File "python", line 1, in <module>
9. NameError: name 'colors' is not defined

Because It makes no sense

1. Complete “Lesson 5: Variables – String Operators” by typing the sample commands in the black area of the IDE.
   1. Explain why the following commands give different results:
      * Color + day \* fishes
      * ( Color + day ) \* fishes
      * Both of them give trace back errors  
        unless you assign values to the words and they give different results because its following BEDMAS
2. Complete “Lesson 5: Variables – Indexes” by typing the sample commands in the black area of the IDE.
   1. What is the index of ‘r’ in “watermelon”?

4

* 1. Write an expression using mynumber to return ‘r’  
     my number equals 5

Watermelon[mynumber – 1]

1. Complete “Lesson 5: Variables – Assignments or Comparisons” by typing the sample commands in the black area of the IDE.
   1. What is the difference between “=” and “==”?

= is assigning values and == is asking a question.

* 1. Create your own mnemonic to remember this difference.

?

1. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.
   1. What doesn’t “friend” + 5 work?

Because you cant add 5 to the friend words because ‘friends’ is a string nad 5 is a int

* 1. Wht is the difference between int and str?

Int means integer which is a whole number while str means strings which are things insdie quotation marks

1. Read through “Lesson 6: Errors – Parts of an Error Message”.
   1. Is “friend” + 5 an example of:
      1. A Syntax Error?
      2. A Runtime Error?
      3. A Logic Error?

It is a logical error

1. Read through “Lesson 6: Errors – Fixing Errors”.
   1. Use the ‘print’ command to print your first name and last name.

Print(‘bob marley’)

1. Complete “Lesson 7: Booleans – Types of Data” by typing the sample commands in the black area of the IDE.
   1. What is the value of: type(“True”) class str
   2. What is the value of: type( True ) class bol
   3. Why is the result different?

Because the first one has quotes and that makes it a string

1. Complete “Lesson 7: Booleans – What Is A Boolean” by typing the sample commands in the black area of the IDE.
   1. Why do you think that having a Boolean data type is important in computer programming?
   2. If you’re making a game, the game has to have options so that’s whre Booleans come in to make it so you can select opetions.
2. Complete “Lesson 7: Booleans – Trying Out Booleans” by typing the sample commands in the black area of the IDE.
   1. Why do you think that there is no Maybe” Boolean data value in computer programming?

Because when your writing a code it has to be either true or false you cant have maybe somethings either true or false a yes or a no unless the computers unsure then itll come back as a error.

**Level 3: Lists & Logic**

Access the Tutorial and start at “Lesson 7: Booleans”

Questions

1. Complete “Lesson 7: Booleans – AND Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. True and True
      2. True and False
      3. False and True
      4. False and False
   2. Explain if there are any other combinations of True / False.
   3. Explain how the AND operator is similar to a math operator and how it is different.

The and operators compare to things that are similar together, but it does not do the math like a Boolean it answers true or false questions.

1. Complete “Lesson 7: Booleans – OR Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. True or True
      2. True or False
      3. False or True
      4. False or False
   2. Explain how the OR operator is similar to the AND operator and how it is different.

what happens when we use the word or between comparisons?

As long as at least one part of the comparison is True, the whole expression is considered True.

Let's think back to our grocery store example:

You can pay with cash OR a credit card OR a check - as long as you have at least one of those things, you can buy your cheese.

1. Complete “Lesson 7: Booleans – NOT Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. not (True or True)
      2. not (True or False)
      3. not (False or True)
      4. not (False or False)
   2. Explain how the combination of the NOT & OR operators is similar to the AND operator by itself and how it is different.
2. Complete “Lesson 7: Booleans – Expressions” by typing the sample commands in the black area of the IDE.
   1. Explain why the following two Python statements give different results.
      1. not (True or True)
      2. not True or True

the brackets will make the difference in this equation.

* 1. Explain why the following two Python statements give the same results.
     1. not (True and True)
     2. not True and True

because they are using the and operators

1. Complete “Lesson 7: Booleans – Practice” by typing the sample commands in the black area of the IDE.
   1. Create three more practice expressions similar to those in the tutorial.
   2. Provide the results for your practice expressions

**not false and false**

Traceback (most recent call last):

File "python", line 1, in <module>

1. NameError: name 'false' is not defined

**not (false and false)**

Traceback (most recent call last):

File "python", line 1, in <module>

NameError: name 'false' is not defined

1. Complete “Lesson 8: Lists – A Collection of Objects” by typing the sample commands in the black area of the IDE.
   1. Create a list of your favorite sports teams.
   2. Assign your list to a variable.
   3. Confirm that your variable and your list are the same.

Jets = (‘raptors,jays,raptors’)

1. Complete “Lesson 8: Lists – List Indexes” by typing the sample commands in the black area of the IDE.
   1. What is the list index of the last team in your list of favorite sports teams.

2

* 1. In the tutorial, the error produced by typing “fruit[3]” is an example of:
     1. A Syntax Error? <
     2. A Runtime Error?
     3. A Logic Error?

1. Complete “Lesson 8: Lists – Practice” and “Lesson 8: Lists – Practice Answers” by typing the sample commands in the black area of the IDE.

NOTE: Starting with Lesson 9 you should use the WHITE area of the IDE for entering example code with multiple statements.

1. Complete “Lesson 9: Logic – Making Decisions” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code to print “Hi Alfred!” based on a decision using numbers

Done

1. Complete “Lesson 9: Logic – Adding A Choice” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code to print your first name or your last name based on a choice (using “else”).

myname = 'boss'

if myname == "bob":

print ('yes')

else: print('no')

1. Complete “Lesson 9: Logic – Adding Many Choices” and “Lesson 9: Logic – Practice” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code and “elif” statements to make a choice using at least 4 of your friends names.

myname = 'boss'

if myname == "bob": print('yes')

elif if myname == 'jake': print('no')

elif if myname == 'bob': print('maybe')

elif if myname == 'todd': printi('kinda')